

Санкт-Петербургский политехнический
университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе
Дисциплина: Телекоммуникационные технологии
Тема: Помехоустойчивое кодирование

Выполнил студент группы 33501/3

(подпись) П.М.Шувалов

Преподаватель

(подпись) Н.В.Богач

Санкт-Петербург
2017

1 Помехоустойчивое кодирование

1.1 Цель

Изучение методов помехоустойчивого кодирования и сравнение их свойств.

1.2 Постановка задачи

- Провести кодирование/декодирование сигнала, полученного с помощью функции `randerr` кодом Хэмминга двумя способами: с помощью встроенных функций `encode/decode`, а также через создание проверочной и генераторной матриц и вычисление синдрома. Оценить корректирующую способность кода.
- Выполнить кодирование/декодирование циклическим кодом, кодом БЧХ, кодом Рида-Соломона. Оценить корректирующую способность кода.

1.3 Теоретические положения

Кодирование передаваемого сообщения позволяет осуществлять его проверку на наличие ошибок при получении, а в некоторых случаях и исправлять их. Данная возможность достигается за счет введения информационной избыточности, что уменьшает удельное количество полезной информации в сообщении.

Значительную долю кодов составляют **блочные коды**. При их применении передаваемое сообщение разбивается на блоки одинаковой длины, после чего каждому блоку сопоставляется код в соответствии с выбранным способом кодирования.

Другая характеристика, позволяющая выделить коды в отдельный класс - цикличность. У кодов этого класса циклическая перестановка букв слова также является кодовым словом.

Код Хэмминга является циклическим самокорректирующимся кодом. Помимо информационных бит в сообщении передается набор контрольных бит, которые вычисляются как сумма по модулю 2 всех информационных бит, кроме одного. Для m контрольных бит максимальное число информационных бит составляет $2^m - n - 1$. Код Хэмминга позволяет обнаружить до двух ошибок при передаче и исправить инверсную передачу одного двоичного разряда.

Коды БЧХ позволяют при необходимости исправлять большее число ошибок в разрядах за счет внесения дополнительной избыточности. Они принадлежат к категории блочных кодов. Частным случаем БЧХ кодов являются коды **Рида-Соломона**, которые работают с недвоичными данными. Их корректирующая способность, соответственно, не ниже, чем у кодов Хэмминга.

Кодировку сообщения производят с помощью генераторной матрицы, домножение на которую столбца создает закодированное сообщение. На приемной стороне сообщение домножается на проверочную матрицу, полученный результат называется синдромом и позволяет определить наличие ошибок и их местоположение, если корректирующая способность кода достаточна.

1.4 Ход работы

В работе разными способами передается сообщение из 4 бит (взято сообщение из соответствующего варианту на контрольной работе: 0111) - с помощью функций encode/decode в коде Хэмминга, с помощью генераторной и проверочной матриц кода Хэмминга, с ручным подсчетом контрольных бит кода Хэмминга, а также с помощью создания полинома циклического кода и встроенными методами кодирования в кодах БЧХ и Рида-Соломона.

Код приведен в листинге.

```
clear all;
clc;
% Сообщение
MESSAGE_LENGTH = 4;
CODE_LENGTH = 7;
message = [0 1 1 1] % сообщение из варианта контрольной работы
%% Метод Хэмминга
% Кодирование/декодирование
fprintf('Hamming encoding with encode/decode');
code = encode(message, CODE_LENGTH, MESSAGE_LENGTH, 'hamming/binary')
dec = decode(code, CODE_LENGTH, MESSAGE_LENGTH, 'hamming/binary')
% Матрицы
fprintf('Hamming encoding with matrices');
[h, g, n, k] = hamngen(CODE_LENGTH-MESSAGE_LENGTH);
code = rem(message*g, ones(1,n).*2)
random_bit_num = 1; % Случайное расположение ошибки
code(random_bit_num) = not(code(random_bit_num)) % Добавление ошибки
syndrome = rem(code*h', ones(1,n-k).*2)

err_place = bi2de(syndrome);
if err_place ~= 0 % Исправление
    fprintf('Error in %i bit', err_place);
    code(err_place) = not(code(err_place))
end
decode_message = [code(4), code(5), code(6), code(7)]

% Хэмминг с проверочными битами
fprintf('Hamming with xors');
code = ham_code(message)
[decode, syndrome] = ham_decode(code)

% Циклический код
fprintf('Cycle code');
pol = cyclpoly(CODE_LENGTH, MESSAGE_LENGTH);
[h_cycle, g_cycle] = cyclgen(CODE_LENGTH, pol);
codecycle = message*g_cycle;
codecycle = rem(codecycle, ones(1, CODE_LENGTH).*2)
decode_message = [code(4), code(5), code(6), code(7)]
```

```

syndrome = rem(codecycle*h_cycle', ones(1, CODE_LENGTH-MESSAGE_LENGTH).*2)

% БЧХ
fprintf('BCH code');
code_bch = comm.BCHEncoder(CODE_LENGTH, MESSAGE_LENGTH);
dec_bch = comm.BCHDecoder(CODE_LENGTH, MESSAGE_LENGTH);
temp = message';
code = step(code_bch, temp(:))';
decode = step(dec_bch, code')'

% Рид-Соломон
fprintf('RS code');
code_rs = comm.RSEncoder(CODE_LENGTH, MESSAGE_LENGTH);
dec_rs = comm.RSDecoder(CODE_LENGTH, MESSAGE_LENGTH);
temp = message';
code = step(code_rs, temp(:))';
decode = step(dec_rs, code')'

```

1.5 Результаты

```
% Кодируемое сообщение
message =

    0    1    1    1

% Код Хэмминга с использованием encode-decode
Hamming encoding with encode/decode
code =

    0    0    1    0    1    1    1

dec =

    0    1    1    1
% Код Хэмминга с использованием матриц
Hamming encoding with matrices
code =

    0    0    1    0    1    1    1

% Добавление ошибки
code =

    1    0    1    0    1    1    1

% Ошибка найдена
syndrome =

    1    0    0

Error in 1 bit
code =

    0    0    1    0    1    1    1

% сообщение декодировано верно
decode_message =

    0    1    1    1
% Хэмминг с проверочными битами
Hamming with xors
code =

    0    1    1    1    0    0    1
```

decode =

0 1 1 1

syndrome =

0 0 0

% Циклический код

Cycle code

codecycle =

0 1 0 0 1 1 1

decode_message =

1 0 0 1

syndrome =

0 0 0

% БЧХ

BCH code

code =

0 1 1 1 0 1 0

decode =

0 1 1 1

% Код Рида-Соломона

RS code

code =

0 1 1 1 3 0 5

decode =

0 1 1 1

1.6 Выводы

В результате выполнения работы были изучены методы помехоустойчивого кодирования и изучены их свойства. Были рассмотрены различные методы кодирования, такие как: код Хэмминга, циклический код, БЧХ код, код Рида-Соломона. Выбор кода всегда зависит от поставленной задачи.